Monitoring Software Health in a Heterogeneous HPC Environment

David R. Montoya, Georgia A. Pedicini, David O. Gunter, Jennis Pruett, Jennifer M. Frye, HPC-4

High-performance computing (HPC) clusters are often just reaching production quality when new hardware is arriving for the next capacity or capability cluster. Newer machines have different system, network, and input/output (I/O) capabilities, or may have a different architecture, as will be the case with the Roadrunner hybrid system. The result is a loss of consistency across generations of hardware and software. Software tools must be available to help manage and plan for a diverse environment with different software requirements and configurations.

In a constantly changing HPC environment, we need a global view of system health attributes. Automated tasks need to check the health of the systems. For example, what products are available or missing from all the relevant segments?

In our current environment, "segments" describe the individual units that make up a portion of the larger machine. On some clusters, different segments provide different functionality and must be viewed as separate machines. Front-end and compile nodes add another dimension of complexity. Take these all into account and we have more than 50 machines to monitor.

In the past few years we have been making plans to better support the changes in HPC environments. We have instituted projects to collect data from across clusters to support managing the software. An initial tool allows for our software environment to be analyzed, assessing consistency across machines. Looking at the complete consolidated module listing (Fig. 1), there will be differences: certain products will not run with a specific processor, or at a kernel release, or library versions are not compatible with a specific release.

This display is a visual approach to the question of software availability across the production environment. Major differences are easily apparent. A nightly collection process also collects details about product versions. Further enhancements will add these details, making it easier to pinpoint minor discrepancies within clusters.

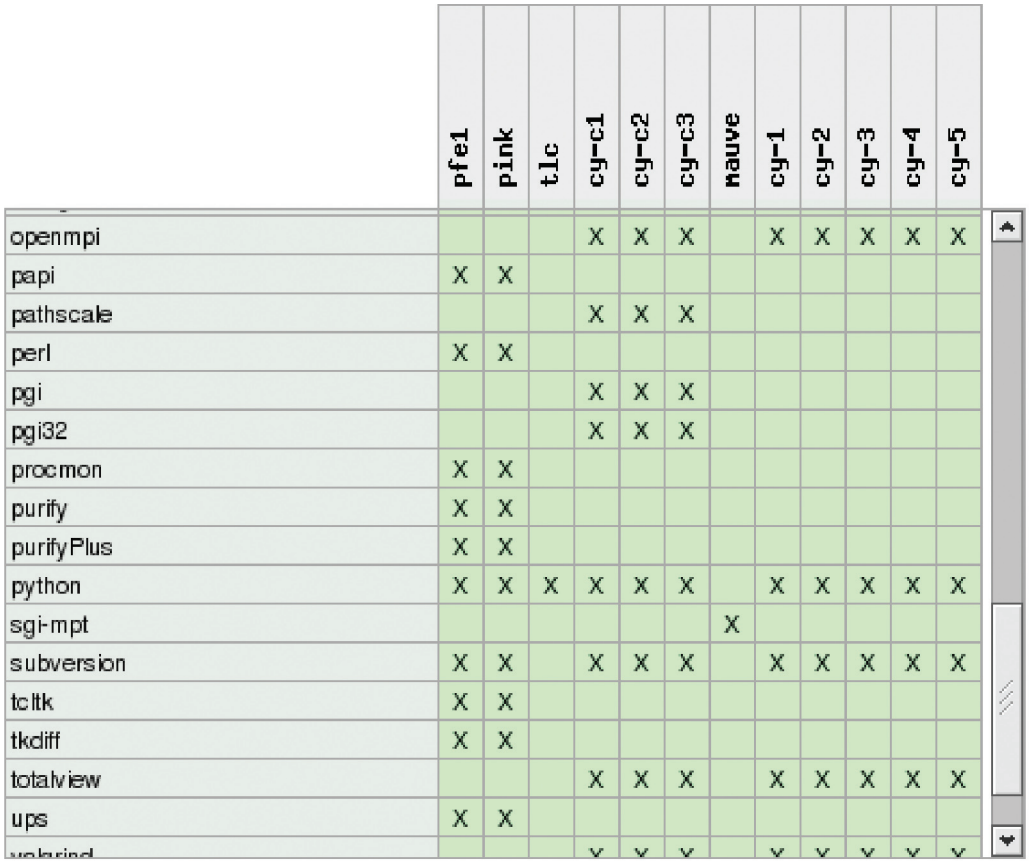
System Health

The capability represented in Fig. 1 is just an initial step to a broader infrastructure to manage the health of the software environment. Two other questions need data: (1) whether the available products are in good working condition, and (2) evaluating usage patterns. Both efforts are underway. A regression-test suite is being added to the data collection framework. "Lite" tests can verify minimal product functionality; more complex tests can be developed to evaluate performance. We anticipate nightly regression runs reporting success or failure. For example, we could highlight if there were any failures for a given product or machine segment. Another capability would log details of the test results. Usage data collection is also being prototyped to collect usage

statistics for users and products. We are adapting the custom support service model to our heterogeneous HPC environment. Tools and data are needed to manage the software environment, to make decisions about which products provide value to the environment, and to manage product lifecycles. Initial steps have been taken, and enhancements are planned. With a focus on efficiency and value added, we are building a sustainable support capability.

For more information, contact David Montoya at dmont@lanl.gov.

Funding Acknowledgements
This research was supported by the NNSA tri-Lab Advanced Simulation and Computing Program.



	pfe1	pink	tlc	cy-c1	cy-c2	cy-c3	nauve	cy-1	cy-2	cy-3	cy-4	cy-5
openmpi				X	X	X		X	X	X	X	X
papi	X	X										
pathscale				X	X	X						
perl	X	X										
pgi				X	X	X						
pgi32				X	X	X						
procmon	X	X										
purify	X	X										
purifyPlus	X	X										
python	X	X	X	X	X	X		X	X	X	X	X
sgi-mpt							X					
subversion	X	X		X	X	X		X	X	X	X	X
tcltk	X	X										
tkdiff	X	X										
totalview				X	X	X		X	X	X	X	X
ups	X	X										
valgrind				X	X	X		X	X	X	X	X

Fig. 1.
A sample segment of software product distribution on the LANL Turquoise network. Tools are listed in the left column and machine segments along the top.